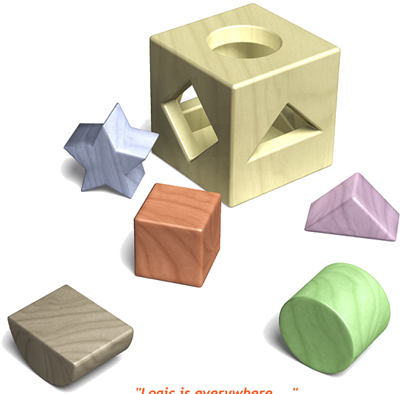
SAT Solving – Systematic Search

Steffen Hölldobler and Norbert Manthey
International Center for Computational Logic
Technische Universität Dresden
Germany

- Truth Tables
- Semantic Trees
- DPLL
- DPLL-NB
- DPLL-CDBL
- GenericCDCL



"Logic is everywhere ..."

Truth Tables

- How can we compute the value of a formula F under all possible interpretations?
- Computing a truth table
  1. Let $m = |\mathcal{S}(F)|$ be the number of subformulae of F
  2. Let $\mathcal{R}_F = \{A \mid A \in \mathcal{R} \text{ and } A \in \mathcal{S}(F)\}$ and $n = |\mathcal{R}_F|$ be the number of propositional variables occurring in F
  3. Form a table $\mathrm{TT}(F)$ with 2^n rows and m columns, where the first n columns are marked by the elements of $\mathcal{R}_F$, the last column is marked by F, and the remaining columns are marked by the other subformulas of F
  4. Fill in the first n columns with $\top$ and $\bot$ as follows: In the first column fill in alternating downwards $\top\bot\top\bot\ldots$, in the second column $\top\top\bot\bot\ldots$, in the third column $\top\top\top\top\bot\bot\bot\bot\ldots$, etc.
  5. Calculate the values in the remaining columns using the known functions on the set of truth values

Some Details

- For a row ζ in $\mathrm{TT}(F)$ we denote by $\zeta(G)$ the truth value in the column marked by the formula $G \in \mathcal{S}(F)$
- With this, step 5 can be reformulated as follows

5. For each row ζ in $\mathrm{TT}(F)$ and for all $F \circ G,\ \neg F \in \mathcal{S}(F) \setminus \mathcal{R}_F$ calculate:

$$\zeta(F \circ G) = \zeta(F) \circ^* \zeta(G) \text{ and } \zeta(\neg F) = \neg^* \zeta(F)$$

Some Observations

- Let I be an interpretation, F a formula,
  $\mathcal{R}_F$ the set of variables occurring in F and $n = |\mathcal{R}_F|$
- If we fix A^I for all $A \in \mathcal{R}_F$, then F^I is uniquely determined
- There are exactly 2^n different possibilities of assigning truth values to $\mathcal{R}_F$
- Each row in TT(F) corresponds precisely to one of these possibilities
- For each interpretation I of the language $\mathcal{L}(\mathcal{R})$
  exists exactly one row ζ_I in TT(F) with $G^I = \zeta_I(G)$ for all $G \in \mathcal{S}(F)$
- For every row ζ in TT(F) exists an interpretation I of the language $\mathcal{L}(\mathcal{R})$
  with $G^I = \zeta(G)$ for all $G \in \mathcal{S}(F)$
  - I is not uniquely determined

Determining Satisfiability using Truth Tables

- F is satisfiable **iff** $\mathrm{TT}(F)$ contains a row ζ with $\zeta(F) = \top$
- F is unsatisfiable **iff** for all rows ζ in $\mathrm{TT}(F)$ we find $\zeta(F) = \bot$
- F is valid **iff** for all rows ζ in $\mathrm{TT}(F)$ we find $\zeta(F) = \top$
- F is falsifiable **iff** $\mathrm{TT}(F)$ contains a row ζ with $\zeta(F) = \bot$

Semantic Trees – Main Characteristics

- Optimization of the truth table method
- Stepwise partitioning of interpretations (through branching)
- Usually conceived for formulas in clausal form
- **Notatation**
  In the sequel nodes are (labeled by) expressions of the form $F :: J$, where
  - F is a formula and
  - J is either a partial interpretation for F, SAT, or CONFLICT

  with the following informal meaning:
  - If J is a partial interpretation, then $F|_J$ is the “remaining” SAT-problem
  - If J is SAT or CONFLICT, then $F|_J$ is undefined
  - $F :: J$ has successor $F :: \mathrm{SAT}$ iff $J \models F$ iff $F|_J = \langle\,\rangle$
  - $F :: J$ has successor $F :: \mathrm{CONFLICT}$ iff $J \not\models F$ iff $[\,] \in F|_J$

Semantic Trees

- **A semantic tree for a CNF-formula F is a binary tree satisfying the following conditions**
  - **The root node is $F :: (\,)$**
  - **If $F :: J$ is a node and $F|_J = \langle\,\rangle$ then it has a successor node $F :: \mathrm{SAT}$**
  - **If $F :: J$ is a node and $[\,] \in F|_J$ then it has a successor node $F :: \mathrm{CONFLICT}$**
  - **If $F :: J$ is a node, $[\,] \notin F|_J$ and $A \in \mathrm{atoms}(F|_J)$ then $F :: J$ has two successor nodes $F :: J, \dot{A}$ and $F :: J, \dot{\overline{A}}$**
- **Note**
  - **The conditions in the three if-statements are mutually exclusive; the corresponding rules (see next slide) do not overlap**
  - **Instead of $F :: J$ we could label the nodes with $F|_J$**
  - **From an implementational point of view it is beneficial to separate F and J Why?**

The ST Calculus

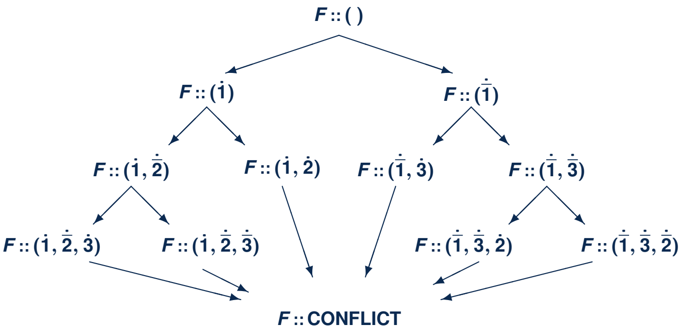
- Given a CNF-formula F
- Computations are initialized by $F :: (\,)$
- The rules of the calculus are

$$
\begin{array}{lllll}
F :: J & \leadsto_{SAT} & F :: \mathrm{SAT} & \text{iff} & F|_J = \langle\,\rangle \\
F :: J & \leadsto_{CONF} & F :: \mathrm{CONFLICT} & \text{iff} & [\,] \in F|_J \\
F :: J & \leadsto_{SPLIT} & F :: J, \dot{A} \mid F :: J, \dot{\overline{A}} & \text{iff} & A \in \mathrm{atoms}(F|_J) \text{ and } [\,] \not\in F|_J
\end{array}
$$

- SPLIT leads to branching
- Computation terminates if
  - a node $F :: \mathrm{SAT}$ is reached in which case F is satisfiable or
  - all leaf nodes are of the form $F :: \mathrm{CONFLICT}$ in which case F is unsatisfiable
- $F :: J \leadsto F' :: J'$
  iff $F :: J \leadsto_{SAT} F' :: J'$ or $F :: J \leadsto_{CONF} F' :: J'$ or $F :: J \leadsto_{SPLIT} F' :: J'$
- $\overset{*}{\leadsto}$ is the reflexive and transitive closure of $\leadsto$

Example

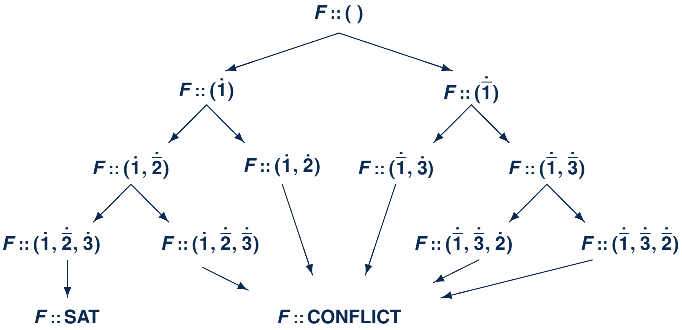
- Let $F = \langle[2, \overline{3}], [2, 3], [\overline{1}, \overline{2}], [1, \overline{3}], [1, \overline{2}, 3]\rangle$ in



- F is unsatisfiable

Another Example

- **Let** $F = \langle [2, 3], [\overline{1}, \overline{2}], [1, \overline{3}], [1, \overline{2}, 3] \rangle$ **in**



- $(\dot{1}, \dot{\overline{2}}, \dot{3}) \models F$

Abstract Reduction Systems

- The ST calculus is an abstract reduction system (see e.g. Baader, Nipkow: Term Rewriting and All That. Cambridge University Press: 1998)
- An abstract reduction system $(\mathcal{R}, \rightarrow)$ is said to be

terminating	iff	there is no infinite descending chain $t_0 \rightarrow t_1 \rightarrow \ldots$
confluent	iff	$t_1 \leftarrow^* t \rightarrow^* t_2$ implies $(\exists t')\, t_1 \rightarrow^* t' \leftarrow^* t_2$
locally confluent	iff	$t_1 \leftarrow t \rightarrow t_2$ implies $(\exists t')\, t_1 \rightarrow^* t' \leftarrow^* t_2$
canonical	iff	is is terminating and confluent

- Newman's Lemma A terminating relation is confluent if it is locally confluent
  Newman: On theories with a combinatorial definition of 'equivalence'. Annals of Mathematics 43(2), 223-243: 1942

ST Termination

- **Theorem** ST is terminating
- **Proof (sketch)**
  - SAT, CONF and SPLIT do not overlap, i.e., at most one of these rules is applicable to a node $F :: J$
  - If SAT or CONF are applied then their only successor nodes $F ::$ SAT and $F ::$ CONFLICT are irreducible
  - We turn to SPLIT
    - atoms(F) is finite
    - SPLIT is applied to $F :: J$ yielding two successor nodes $F :: J, \dot{A}$ and $F :: J, \dot{\overline{A}}$ if $A \in$ atoms($F|_J$)
    - $A \not\in$ atoms($F|_{(J,\dot{A})}$) $\cup$ atoms($F|_{(J,\dot{\overline{A}})}$)
    - There are no infinite sequences of SPLIT qed

ST Confluency – Preliminaries

- We assume that nodes are labeled by $F|_J$ instead of $F :: J$
- **Observations**
  - $F|_{J,L_1,L_2} = F|_{J,L_2,L_1}$
  - $F|_J = F|_{J,A} = F|_{J,\overline{A}}$ if $A \not\in$ atoms$(F|_J)$ and neither $A \in J$ nor $\overline{A} \in J$

Example

- Let $F = \langle [1, 2], [1, 3], [4, 5] \rangle$

$$F$$

$$F|_{\dot{2}} = \langle [1, 3], [4, 5] \rangle \qquad F|_{\dot{\overline{2}}} = \langle [1], [1, 3], [4, 5] \rangle$$

$$F|_{\dot{2},\dot{1}} = \langle [4, 5] \rangle \qquad F|_{\dot{2},\dot{\overline{1}}} = \langle [3], [4, 5] \rangle \qquad F|_{\dot{\overline{2}},\dot{1}} = \langle [4, 5] \rangle \qquad F|_{\dot{\overline{2}},\dot{\overline{1}}} = \langle [\,], [3], [4, 5] \rangle$$

- The set of leaves is $\{\langle [4, 5] \rangle, \langle [3], [4, 5] \rangle, \langle [\,], [3], [4, 5] \rangle\}$

Example – Continued

- Let $F = \langle [1,2], [1,3], [4,5] \rangle$

$$F$$

$$F|_{\dot{1}} = \langle [4,5] \rangle = F|_{\dot{2},\dot{1}} = F|_{\dot{\overline{2}},\dot{1}}$$

$$F|_{\dot{\overline{1}}} = \langle [2], [3], [4,5] \rangle$$

$$F|_{\dot{\overline{1}},\dot{2}} = \langle [3], [4,5] \rangle = F|_{\dot{2},\dot{\overline{1}}}$$

$$F|_{\dot{\overline{1}},\dot{\overline{2}}} = \langle [\,], [3], [4,5] \rangle = F|_{\dot{\overline{2}},\dot{\overline{1}}}$$

- The set of leaves is $\{\langle [4,5] \rangle, \langle [3], [4,5] \rangle, \langle [\,], [3], [4,5] \rangle\}$, which is identical to the set of leaves of the previous tree
- Two trees with identical root are similar if they have identical sets of leaves

ST Confluency

- **Proposition** ST is confluent (modulo similarity of trees)
- **Proof (sketch)** Because ST is terminating and, thus, Newman's Lemma is applicable, it suffices to show that ST is locally confluent
  - Because SAT, CONF, and SPLIT do not overlap, the only possible overlap is between two different instance of SPLIT applicable to some node $F|_J$
  - Let $F|_{J,\dot{A}}$ and $F|_{J,\dot{\overline{A}}}$ as well as $F|_{J,\dot{B}}$ and $F|_{J,\dot{\overline{B}}}$ be the respective extensions of $F|_J$, where $A \neq B$
  - If $B \in \text{atoms}(F|_{J,\dot{A}})$ then SPLIT can be applied to $F|_{J,\dot{A}}$ yielding $F|_{J,\dot{A},\dot{B}}$ and $F|_{J,\dot{A},\dot{\overline{B}}}$; otherwise, $F|_{J,\dot{A}} = F|_{J,\dot{A},\dot{B}} = F|_{J,\dot{A},\dot{\overline{B}}}$
  - Similar arguments can be made for the remaining three cases
  - Because literals can be swapped in an interpretation, the two trees rooted in $F|_J$ and generated by the two different inital applications of SPLIT are similar

qed

A Comment

- $[\,] \not\in F|_J$ may be omitted
  - in the last condition of the definition of a semantic tree and, consequently,
  - in the definition of SPLIT
- Hence,
  - CONF and SPLIT overlap
  - However, ST is still confluent
  - One can show that CONF is a simplification rule
  - Thus, CONF should always be applied first
    and no alternatives need to be considered

ST Soundness

- **Lemma** Suppose $F :: J \rightsquigarrow_{SPLIT} F :: J, \dot{A} \mid F :: J, \dot{\overline{A}}$. Then, $F|_J$ is satisfiable **iff** either $F|_{(J,\dot{A})}$ or $F|_{(J,\dot{\overline{A}})}$ is satisfiable
- **Proof** $\rightsquigarrow$ Exercise
- **Theorem** ST is sound
- **Proof (sketch)**
  - To show if $F :: (\,) \overset{*}{\rightsquigarrow} F :: SAT$ then the CNF-formula F is satisfiable
  - Suppose $F :: (\,) \overset{*}{\rightsquigarrow} F :: SAT$
  - $F :: SAT$ is generated iff its parent node is $F :: J$ and $F|_J = \langle\,\rangle$
  - $\langle\,\rangle$ is satisfiable
  - By induction on the length of the given derivation and using the above mentioned lemma we can show that F is satisfiable **qed**
- **Exercise** Complete the proof of the Theorem

ST Completeness

- **Corollary** Suppose $F :: J \leadsto_{SPLIT} F :: J, \dot{A} \mid F :: J, \dot{\overline{A}}$. Then, $F|_J$ is unsatisfiable **iff** $F|_{(J,\dot{A})}$ and $F|_{(J,\dot{\overline{A}})}$ are unsatisfiable
- **Proof** Follows from the previous lemma by negating both sides of the equivalence **qed**
- **Theorem** ST is complete
- **Proof (sketch)**
  - To show if a CNF-formula F is satisfiable, then $F :: (\,) \overset{*}{\leadsto} F :: \mathrm{SAT}$
  - Suppose F is satisfiable, but $F :: (\,) \overset{*}{\not\leadsto} F :: \mathrm{SAT}$
  - Because ST is terminating, all leaf nodes are of the form $F :: \mathrm{CONFLICT}$
  - For all leaf nodes we find J such that $F :: J \leadsto_{CONF} F :: \mathrm{CONFLICT}$ and $[\,] \in F|_J$
  - By induction on the lenght of the given derivation and using the above mentioned corollary we learn that F is unsatisfiable **qed**

Relationship to the Truth Table Method

- **Each branch of a semantic tree corresponds to rows in the truth table**
- **Different branches correspond to different rows in the truth table**
- **Advantage over the truth table method**
  - **Leaf nodes $F :: \text{CONFLICT}$ and $F :: \text{SAT}$ may be reached even if not all members of atoms(F) have been assigned to truth values**

Controlling the Generation of Semantic Trees

- **Which leaf node $F :: J$ shall be selected?**
- **Which atom A shall be selected in an application of SPLIT?**
- **Which branch shall be investigated first after an application of SPLIT?**
- **What about redundancies?**

DPLL

- **This section is based on**
  - Davis, Putnam: A Computing Procedure for Quantification Theorem
    Journal of the ACM 7, 201-215: 1960
  - Davis, Logemann, Loveland: A Machine Program for Theorem Provin.
    Communications of the ACM 5, 394-397: 1962
- **DPLL is an acronym for the authors**
- **The DPLL method was originally specified to show unsatisfiability**
- **Here, we present a version for showing satisfiability**
  leading to an improved algorithm for the generation of semantic trees
- **We consider clauses as sets**
  - **There are no multiple occurrences of literals in a clause**
  - **An implementation has to guarantee this!**

Simplification Rules

- Consider a CNF-formula F
- Consider rules which yield F' such that $F' \equiv F$ and F' is "simpler" than F
- Such rules can be applied at any time.
- They are often called simplification rules
- Here TAUT and SUBS

TAUT: Tautological Clauses

- **Definition**
  A clause is a tautology iff it contains a complementary pair of literals
- **Proposition**
  Tautologies can be deleted while preserving semantic equivalence,
  i.e., $F, C \equiv F$ if C is a tautology.

$$\langle [1, \overline{2}, 2, 3, 4, \overline{7}], [5, \overline{6}] \rangle \equiv \langle [5, \overline{6}] \rangle$$

- $F, C :: J \leadsto_{TAUT} F :: J$ iff C is a tautology
- Applicable
  - in the initialization phase
  - whenever a new clause is generated, e.g., by resolution
- TAUT reduces the number of clauses in a formula

SUBS: Subsumption

- **Definition** C_1 subsumes C_2 iff $C_1 \subseteq C_2$
- **Proposition**
  Subsumed clauses can be deleted while preserving semantic equivalence, i.e., $F, C \equiv F$ if there exists $C' \in F$ with $C' \subseteq C$

$$\langle [2, \overline{3}], [\overline{2}], [1, 2, \overline{3}, \overline{4}] \rangle \equiv \langle [2, \overline{3}], [\overline{2}] \rangle$$

- $F, C :: J \leadsto_{SUBS} F :: J$ iff there exists $C' \in F$ such that $C' \subseteq C$
- Applicable
  - in the initialization phase
  - whenever a new clause is generated, e.g., by resolution
- SUBS reduces the number of clauses in a formula
- Some Questions
  - How complex is the removal of subsumed clauses?
  - Are there forms of subsumption which are less costly?

Remaining Rules

- SAT, SPLIT and CONFLICT as in the ST calculus
- UNIT as a special variant of SPLIT
- PURE

UNIT

- Let $F :: J$ be a node in the computation of a semantic tree for F
- Suppose SPLIT was applied yielding the new nodes $F :: J, \dot{L}$ and $F :: J, \dot{\overline{L}}$
  - If $[L] \in F|_J$ then $[\,] \in F|_{(J,\dot{\overline{L}})}$ and, thus, $F :: J, \dot{\overline{L}} \leadsto_{CONF} F :: \text{CONFLICT}$
  - If $[\overline{L}] \in F|_J$ then $[\,] \in F|_{(J,\dot{L})}$ and, thus, $F :: J, \dot{L} \leadsto_{CONF} F :: \text{CONFLICT}$
- Hence, unit clauses should eagerly trigger SPLITs
- $F :: J \leadsto_{UNIT} F :: J, L$ iff $[L] \in F|_J$
- L is a propagation variable
- Applicable
  - in the initialization phase
  - whenever a new clause is generated, e.g., by resolution
  - whenever literals are deleted from a clause, e.g., by UNIT
- How complex is the application of UNIT?

PURE

- **Definition** A literal $L \in \text{lits}(F)$ is **pure** **iff** $\overline{L} \notin \text{lits}(F)$
- Clauses containing a pure L are satisfied by any interpretation containing L
- Interpretations containing $\overline{L}$ need not be considered
- $F :: J \leadsto_{PURE} F :: J, L$ **iff** there exists $L \in \text{lits}(F|_J)$ which is pure in $F|_J$.
- L is a propagation variable
- Applicable
  - in the initialization phase
  - whenever clauses have been deleted, e.g., by SUBS or TAUT

An Example

$$\langle [1,2], [2,\overline{3}], [1,2,\overline{3}], [1,4,\overline{4}], [\overline{2},\overline{3},4], [\overline{1},3], [\overline{4}] \rangle :: (\,)$$

$$\big\downarrow \textit{SUBS}$$

$$\langle [1,2], [2,\overline{3}], [1,4,\overline{4}], [\overline{2},\overline{3},4], [\overline{1},3], [\overline{4}] \rangle :: (\,)$$

$$\big\downarrow \textit{TAUT}$$

$$F :: (\,) \text{ where } F = \langle [1,2], [2,\overline{3}], [\overline{2},\overline{3},4], [\overline{1},3], [\overline{4}] \rangle$$

$$\big\downarrow \textit{UNIT}$$

$$F :: (\overline{4})$$

$$\textit{SPLIT}$$

$$F :: (\overline{4}, \dot{1}) \xrightarrow{\textit{UNIT}} F :: (\overline{4}, \dot{1}, 3) \xrightarrow{\textit{UNIT}} F :: (\overline{4}, \dot{1}, 3, 2) \xrightarrow{\textit{CONF}} F :: \text{CONFLICT}$$

$$F :: (\overline{4}, \dot{\overline{1}}) \xrightarrow{\textit{UNIT}} F :: (\overline{4}, \dot{\overline{1}}, 2) \xrightarrow{\textit{UNIT}} F :: (\overline{4}, \dot{\overline{1}}, 2, \overline{3}) \xrightarrow{\textit{SAT}} F :: \text{SAT}$$

The DPLL Calculus

- Given a CNF-formula F
- The computation is initialized by $F :: (\,)$
- The rules of the calculus are
  SAT, CONFLICT, SPLIT, TAUT, SUBS, UNIT and PURE
- Computation terminates if
  - a node $F' :: \text{SAT}$ is reached in which case F is satisfiable or
  - all leaf nodes are of the form $F' :: \text{CONFLICT}$ in which case F is unsatisfiable
- **Note**
  - TAUT and SUBS may still be applicable to $F' :: \text{CONFLICT}$
  - However, TAUT and SUBS can only be applied finitely many times
    because in each application a clause from F' is removed and F' is finite

UNIT and PURE Revisited

- **Proposition** UNIT and PURE are satisfiability preserving, i.e.,
  - Suppose $F :: J \leadsto_{UNIT} F :: J, L$. Then,
    $F|_J$ is satisfiable **iff** $F|_{J,L}$ is satisfiable
  - Suppose $F :: J \leadsto_{PURE} F :: J, L$. Then,
    $F|_J$ is satisfiable **iff** $F|_{J,L}$ is satisfiable
- **Exercise** Prove the proposition
- **Note** Whenever UNIT or PURE is applied to $F :: J$ yielding $F :: J, L$ then
  - $L \not\in \text{lits}(F|_{J,L})$
  - $\text{atoms}(L) \not\subseteq \text{atoms}(F|_{J,L})$
  - $\text{lits}(F|_{J,L}) \cup \{L\} \subseteq \text{lits}(F|_J)$
- **Exercise** Give examples for the application of UNIT and PURE where the subset relation is proper

DPLL Termination

- **Theorem** DPLL is terminating
- **Proof (sketch)** The theorem follows from the following observations:
  - SAT: the node cannot be further extended
  - CONFLICT: the node will not be further extended
  - TAUT, SUBS, UNIT, PURE: the number of clauses occurring in F decreases
  - SPLIT:
    - the number of clauses does not increase
    - the number of atoms occurring in $F|_J$ decreases qed
- **Exercise** Complete the proof

DPLL Confluency

- **Claim** DPLL is confluent
- **Some Consequences**
  - The inference rules can be applied in any order
  - Starting with $F :: ()$ we can first apply TAUT and SUBS as often as possible
  - Thereafter, TAUT and SUBS will not be applicable anymore
  - We delay applications of SPLIT as long as possible, i.e., if TAUT and SUBS are no longer applicable, we apply PURE and UNIT eagerly

DPLL Soundness

- **Theorem** DPLL is sound
- **Proof** Follows from the corresponding result of the GenericCDCL calculus

DPLL Completeness

- **Theorem** DPLL is complete
- **Proof (sketch)**
  The proof is in analogy to the proof of the completeness of the ST calculus
  - Recall that
    - TAUT and SUBS are simplification rules and
    - PURE and UNIT are satisfiability preserving
  - Hence, if $F :: J \leadsto F' :: J'$, where $\leadsto \in \{\leadsto_{TAUT}, \leadsto_{UNIT}, \leadsto_{SUBS}, \leadsto_{PURE}\}$,
    then the following holds: If $F'|_{J'}$ is unsatisfiable, so is $F|_J$.
  - The remaining steps are similar to those in the proof of the
    completeness of the ST calculus except that
    between two splits TAUT, SUBS, PURE, and UNIT may be applied qed
- **Exercise** Complete the proof

Naive Backtracking

- The ST and the DPLL calculus are branching due to SPLIT
- We would like to linearize DPLL and, thereby, ST
- TAUT and SUBS simplify the formula, SAT is a termination rule
- UNIT and PURE are satisfiability preserving and add propagation literals
- SPLIT is replaced by
  - $F :: J \quad \leadsto_{DECIDE} \quad F :: J, \dot{L}$ iff $[\,] \not\in F|_J$ and $L \in \text{atoms}(F|_J) \cup \overline{\text{atoms}(F|_J)}$
- If $[\,] \in F|_J$ then J may or may not contain decision literals
  - $F :: J \quad \leadsto_{UNSAT} \quad F :: \text{UNSAT}$
    iff $[\,] \in F|_J$ and J does not contain a decision literal
  - $F :: J, \dot{L}, P \quad \leadsto_{NB} \quad F :: J, \overline{L}$ iff $[\,] \in F|_{J,\dot{L},P}$, where
    - P is a sequence of propagation literals
    - $\dot{L}$ is the decision literal with the highest level in $J, \dot{L}, P$
    - $\overline{L}$ is a propagation literal
    - NB is called naive backtracking

A Note on PURE

- **Each application of PURE can be replaced by DECIDE, in which case the pure literal *L* used by PURE becomes a decision literal**
- **In most of the literature and almost all systems PURE is not considered**
- **We would like to keep it in the moment as we do not fully understand why an implementation of PURE is so costly or why a particular result is affected by PURE**
- **If, however, one of the methods and techniques presented in the sequel causes a problem due to the fact that PURE adds *L* as a propagation literal to the current partial interpretation then PURE shall be replaced by DECIDE**

The Previous Example Revisited

$$\langle [1,2],[2,\overline{3}],[1,2,\overline{3}],[1,4,\overline{4}],[\overline{2},\overline{3},4],[\overline{1},3],[\overline{4}]\rangle :: (\,)$$

$$\begin{array}{lll}
\leadsto_{SUBS} & \langle [1,2],[2,\overline{3}],[1,4,\overline{4}],[\overline{2},\overline{3},4],[\overline{1},3],[\overline{4}]\rangle :: (\,) & \\
\leadsto_{TAUT} & F :: (\,) & \text{where } F = \langle [1,2],[2,\overline{3}],[\overline{2},\overline{3},4],[\overline{1},3],[\overline{4}]\rangle \\
\leadsto_{UNIT} & F :: (\overline{4}) & (F|_{(\overline{4})} = \langle [1,2],[2,\overline{3}],[\overline{2},\overline{3}],[\overline{1},3]\rangle) \\
\leadsto_{DECIDE} & F :: (\overline{4},\dot{1}) & (F|_{(\overline{4},1)} = \langle [2,\overline{3}],[\overline{2},\overline{3}],[3]\rangle) \\
\leadsto_{UNIT} & F :: (\overline{4},\dot{1},3) & (F|_{(\overline{4},1,3)} = \langle [2],[\overline{2}]\rangle) \\
\leadsto_{UNIT} & F :: (\overline{4},\dot{1},3,2) & (F|_{(\overline{4},1,3,2)} = \langle [\,]\rangle) \\
\leadsto_{NB} & F :: (\overline{4},\overline{1}) & (F|_{(\overline{4},\overline{1})} = \langle [2],[2,\overline{3}],[\overline{2},\overline{3}]\rangle) \\
\leadsto_{UNIT} & F :: (\overline{4},\overline{1},2) & (F|_{(\overline{4},\overline{1},2)} = \langle [\overline{3}]\rangle) \\
\leadsto_{UNIT} & F :: (\overline{4},\overline{1},2,\overline{3}) & (F|_{(\overline{4},\overline{1},2,\overline{3})} = \langle\,\rangle) \\
\leadsto_{SAT} & F :: \text{SAT} &
\end{array}$$

Another Example Revisited

$F :: (\,)$ where $F = \langle [2, \overline{3}], [2, 3], [\overline{1}, \overline{2}], [1, \overline{3}], [1, \overline{2}, 3] \rangle$

$\leadsto_{DECIDE}$	$F :: (\dot{1})$	$(F\|_{(1)} = \langle [2, \overline{3}], [2, 3], [\overline{2}] \rangle)$
$\leadsto_{UNIT}$	$F :: (\dot{1}, \overline{2})$	$F\|_{(1,\overline{2})} = \langle [\overline{3}], [3] \rangle)$
$\leadsto_{UNIT}$	$F :: (\dot{1}, \overline{2}, \overline{3})$	$(F\|_{(1,\overline{2},\overline{3})} = \langle [\,] \rangle)$
$\leadsto_{NB}$	$F :: (\overline{1})$	$(F\|_{(\overline{1})} = \langle [2, \overline{3}], [2, 3], [\overline{3}], [\overline{2}, 3] \rangle)$
$\leadsto_{UNIT}$	$F :: (\overline{1}, \overline{3})$	$(F\|_{(\overline{1},\overline{3})} = \langle [2], [\overline{2}] \rangle)$
$\leadsto_{UNIT}$	$F :: (\overline{1}, \overline{3}, 2)$	$(F\|_{(\overline{1},\overline{3},2)} = \langle [\,] \rangle)$
$\leadsto_{UNSAT}$	$F ::$ UNSAT	

The DPLL-NB Calculus

- Given a CNF-formula F
- The computation is initialized by $F :: (\,)$
- The rules of the calculus are
  SAT, UNSAT, DECIDE, NB, TAUT, SUBS, UNIT and PURE
- Computation terminates if
  - a node $F' ::$ SAT is reached in which case F is satisfiable or
  - a node $F' ::$ UNSAT is reached in which case F is unsatisfiable
- **Note** In DPLL-NB and, in particular, in $F :: J$,
  J may be a partial interpretation, SAT or UNSAT

DPLL-NB – Results

- **Theorem** DPLL-NB is terminating, sound, and complete
- **Proof (sketch)**
  - Termination and soundness follow from corresponding results for the GenericCDCL calculus, which will be presented later in the lecture
  - Completeness
    - DPLL is complete
    - The search space is finite
    - NB specifies just a specific order of traversing this space qed

Heuristics

- Whenever DECIDE is applied to $F :: J$ in the following examples, then
  - the smallest atom A occurring in $F|_J$ is selected
- Whenever UNIT is applied to $F :: J$ in the following examples, then
  - the leftmost unit clause occurring in $F|_J$ is selected

Backtracking and Redundancies (1)

	$F :: (\,)$ where $F = \langle[\overline{1},\overline{2},\overline{3}],[\overline{2},4],[2,4],[\overline{5},6],[\overline{1},\overline{5},\overline{6}],[5,7],[\overline{1},5,\overline{7}],[1,3]\rangle$	
$\leadsto$ *DECIDE*	$F :: (\dot{1})$	$(F\|_{(1)} = \langle[\overline{2},\overline{3}],[\overline{2},4],[2,4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$
$\leadsto$ *DECIDE*	$F :: (\dot{1},\dot{2})$	$(F\|_{(1,2)} = \langle[\overline{3}],[4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\dot{2},\overline{3})$	$(F\|_{(1,2,\overline{3})} = \langle[4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\dot{2},\overline{3},4)$	$(F\|_{(1,2,\overline{3},4)} = \langle[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$
$\leadsto$ *DECIDE*	$F :: (\dot{1},\dot{2},\overline{3},4,\dot{5})$	$(F\|_{(1,2,\overline{3},4,5)} = \langle[6],[\overline{6}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\dot{2},\overline{3},4,\dot{5},6)$	$(F\|_{(1,2,\overline{3},4,5,6)} = \langle[\,]\rangle)$
$\leadsto$ *NB*	$F :: (\dot{1},\dot{2},\overline{3},4,\overline{5})$	$(F\|(\dot{1},\dot{2},\overline{3},4,\overline{5}) = \langle[7],[\overline{7}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\dot{2},\overline{3},4,\overline{5},7)$	$(F\|_{(\dot{1},\dot{2},\overline{3},4,\overline{5},7)} = \langle[\,]\rangle)$
$\leadsto$ *NB*	$F :: (\dot{1},\overline{2})$	$(F\|_{(1,\overline{2})} = \langle[4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\overline{2},4)$	$(F\|_{(\dot{1},\overline{2},4)} = \langle[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$
$\leadsto$ *DECIDE*	$F :: (\dot{1},\overline{2},4,\dot{5})$	$(F\|_{(\dot{1},\overline{2},4,\dot{5})} = \langle[6],[\overline{6}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\overline{2},4,\dot{5},6)$	$(F\|_{(\dot{1},\overline{2},4,\dot{5},6)} = \langle[\,]\rangle)$
$\leadsto$ *NB*	$F :: (\dot{1},\overline{2},4,\overline{5})$	$(F\|_{(\dot{1},\overline{2},4,\overline{5})} = \langle[7],[\overline{7}]\rangle)$
$\leadsto$ *UNIT*	$F :: (\dot{1},\overline{2},4,\overline{5},7)$	$(F\|_{(\dot{1},\overline{2},4,\overline{5},7)} = \langle[\,]\rangle)$
$\leadsto$ *NB*	$F :: (\overline{1})$	$(F\|_{(\overline{1})} = \langle[\overline{2},4],[2,4],[\overline{5},6],[5,7],[3]\rangle)$
$\leadsto$ *UNIT*	$F :: (\overline{1},3)$	$(F\|_{(\overline{1},3)} = \langle[\overline{2},4],[2,4],[\overline{5},6],[5,7]\rangle)$
$\leadsto$ *PURE*	$F :: (\overline{1},3,7)$	$(F\|_{(\overline{1},3,7)} = \langle[\overline{2},4],[2,4],[\overline{5},6]\rangle)$
$\leadsto$ *PURE*	$F :: (\overline{1},3,7,\overline{5})$	$(F\|_{(\overline{1},3,7,\overline{5})} = \langle[\overline{2},4],[2,4]\rangle)$
$\leadsto$ *PURE*	$F :: (\overline{1},3,7,\overline{5},4)$	$(F\|_{(\overline{1},3,7,\overline{5},4)} = \langle\rangle)$
$\leadsto$ *SAT*	$F :: \mathrm{SAT}$	

Conflict Analysis

- In the previous example the following clauses triggered UNIT propagation

$$\begin{array}{lcll} C_1 & = & [\overline{1}, \overline{2}, \overline{3}] & (C_1|_{(1,2)} = [\overline{3}]) \\ C_2 & = & [\overline{2}, 4] & (C_2|_{(1,2)} = [4]) \\ C_3 & = & [\overline{5}, 6] & (C_3|_{(1,2,\overline{3},4,5)}) = [6]) \end{array}$$

Subsequently, the clause $C = [\overline{1}, \overline{5}, \overline{6}]$ became empty and caused a conflict

- We can find the following (linear) resolution derivation from C wrt $\{C_1, C_2, C_2\}$

$$C_4 \quad = \quad [\overline{1}, \overline{5}] \qquad (\text{res}(C, C_3))$$

- **Note**
  - Resolvents can be added while preserving semantic equivalence
  - $[\overline{1}, \overline{5}]|_{(1)} = [\overline{5}]$

Backtracking and Redundancies (2)

$F :: (\,)$ where $F = \langle[\overline{1},\overline{2},\overline{3}],[\overline{2},4],[2,4],[\overline{5},6],[\overline{1},\overline{5},\overline{6}],[5,7],[\overline{1},5,\overline{7}],[1,3]\rangle$

$\leadsto_{DECIDE}$ $F :: (\dot{1})$ $\quad (F|_{(1)} = \langle[\overline{2},\overline{3}],[\overline{2},4],[2,4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$

$\leadsto_{DECIDE}$ $F :: (\dot{1},\dot{2})$ $\quad (F|_{(1,2)} = \langle[\overline{3}],[4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$

$\leadsto_{UNIT}$ $F :: (\dot{1},\dot{2},\overline{3})$ $\quad (F|_{(1,2,\overline{3})} = \langle[4],[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$

$\leadsto_{UNIT}$ $F :: (\dot{1},\dot{2},\overline{3},4)$ $\quad (F|_{(1,2,\overline{3},4)} = \langle[\overline{5},6],[\overline{5},\overline{6}],[5,7],[5,\overline{7}]\rangle)$

$\leadsto_{DECIDE}$ $F :: (\dot{1},\dot{2},\overline{3},4,\dot{5})$ $\quad (F|_{(1,2,\overline{3},4,5)} = \langle[6],[\overline{6}]\rangle)$

$\leadsto_{UNIT}$ $F :: (\dot{1},\dot{2},\overline{3},4,\dot{5},6)$ $\quad (F|_{(1,2,\overline{3},4,5,6)} = \langle[\,]\rangle)$

$\leadsto_{LEARN}$ $F,[\overline{1},\overline{5}] :: (\dot{1},\dot{2},\overline{3},4,\dot{5},6)$

$\leadsto_{BACK}$ $F,[\overline{1},\overline{5}] :: (\dot{1})$

$\leadsto_{UNIT}$ $F,[\overline{1},\overline{5}] :: (\dot{1},\overline{5})$ $\quad ((F,[\overline{1},\overline{5}])|_{(1,\overline{5})} = \langle[\overline{2},\overline{3}],[\overline{2},4],[2,4],[7],[\overline{7}]\rangle)$

$\leadsto_{CDBL}$ $F,[\overline{1},\overline{5}] :: (\dot{1},\overline{5})$ $\quad ((F,[\overline{1},\overline{5}])|_{(1,\overline{5})} = \langle[\overline{2},\overline{3}],[\overline{2},4],[2,4],[7],[\overline{7}]\rangle)$

$\leadsto_{UNIT}$ $F,[\overline{1},\overline{5}] :: (\dot{1},\overline{5},7)$ $\quad ((F,[\overline{1},\overline{5}])|_{(1,\overline{5},7)} = \langle[\overline{2},\overline{3}],[\overline{2},4],[2,4],[\,]\rangle)$

Conflict Analysis (2)

- The following clauses triggered UNIT propagation:

$$\begin{array}{lcll} C_1 & = & [5, 7] & (C_1|_{(1,\overline{5})} = [7]) \\ C_2 & = & [\overline{1}, \overline{5}] & (C_2|_{(1)} = [\overline{5}]) \end{array}$$

- The new conflict was caused by $C = [\overline{1}, 5, \overline{7}]$
- We can find the following (linear) resolution derivation from C wrt $\{C_1, C_3\}$:

$$\begin{array}{lcll} C_3 & = & [\overline{1}, 5] & (\text{res}(C, C_1)) \\ C_4 & = & [\overline{1}] & (\text{res}(C_3, C_2)) \end{array}$$

- Note
  - A unit clause can be added
  - This clause should be considered at the start
  - $[\overline{1}]$ subsumes $[\overline{1}, \overline{5}]$

Backtracking and Redundancies (3)

$$F :: (\,) \text{ where } F = \langle [\overline{1}, \overline{2}, \overline{3}], [\overline{2}, 4], [2, 4], [\overline{5}, 6], [\overline{1}, \overline{5}, \overline{6}], [5, 7], [\overline{1}, 5, \overline{7}], [1, 3] \rangle$$

$\leadsto_{\mathit{DECIDE}}$	$F :: (\dot{1})$	$(F\vert_{(1)} = \langle [\overline{2}, \overline{3}], [\overline{2}, 4], [2, 4], [\overline{5}, 6], [\overline{5}, \overline{6}], [5, 7], [5, \overline{7}] \rangle)$
$\leadsto_{\mathit{DECIDE}}$	$F :: (\dot{1}, \dot{2})$	$(F\vert_{(1,2)} = \langle [\overline{3}], [4], [\overline{5}, 6], [\overline{5}, \overline{6}], [5, 7], [5, \overline{7}] \rangle)$
$\leadsto_{\mathit{UNIT}}$	$F :: (\dot{1}, \dot{2}, \overline{3})$	$(F\vert_{(1,2,\overline{3})} = \langle [4], [\overline{5}, 6], [\overline{5}, \overline{6}], [5, 7], [5, \overline{7}] \rangle)$
$\leadsto_{\mathit{UNIT}}$	$F :: (\dot{1}, \dot{2}, \overline{3}, 4)$	$(F\vert_{(1,2,\overline{3},4)} = \langle [\overline{5}, 6], [\overline{5}, \overline{6}], [5, 7], [5, \overline{7}] \rangle)$
$\leadsto_{\mathit{DECIDE}}$	$F :: (\dot{1}, \dot{2}, \overline{3}, 4, \dot{5})$	$(F\vert_{(1,2,\overline{3},4,5)} = \langle [6], [\overline{6}] \rangle)$
$\leadsto_{\mathit{UNIT}}$	$F :: (\dot{1}, \dot{2}, \overline{3}, 4, \dot{5}, 6)$	$(F\vert_{(1,2,\overline{3},4,5,6)} = \langle [\,] \rangle)$
$\leadsto_{\mathit{CDBL}}$	$F, [\overline{1}, \overline{5}] :: (\dot{1}, \overline{5})$	$((F, [\overline{1}, \overline{5}])\vert_{(1,\overline{5})} = \langle [\overline{2}, \overline{3}], [\overline{2}, 4], [2, 4], [7], [\overline{7}] \rangle)$
$\leadsto_{\mathit{UNIT}}$	$F, [\overline{1}, \overline{5}] :: (\dot{1}, \overline{5}, 7)$	$((F, [\overline{1}, \overline{5}])\vert_{(1,\overline{5},7)} = \langle [\overline{2}, \overline{3}], [\overline{2}, 4], [2, 4], [\,] \rangle)$
$\leadsto_{\mathit{CDBL}}$	$F, [\overline{1}, \overline{5}], [\overline{1}] :: (\overline{1})$	$((F, [\overline{1}, \overline{5}], [\overline{1}])\vert_{(\overline{1})} = \langle [\overline{2}, 4], [2, 4], [\overline{5}, 6], [5, 7], [3] \rangle)$
$\leadsto_{\mathit{UNIT}}$	$F, [\overline{1}, \overline{5}], [\overline{1}] :: (\overline{1}, 3)$	$((F, [\overline{1}, \overline{5}], [\overline{1}])\vert_{(\overline{1},3)} = \langle [\overline{2}, 4], [2, 4], [\overline{5}, 6], [5, 7] \rangle)$
$\overset{3}{\leadsto}_{\mathit{PURE}}$	$F, [\overline{1}, \overline{5}], [\overline{1}] :: (\overline{1}, 3, 4, 6, 7)$	$((F, [\overline{1}, \overline{5}], [\overline{1}])\vert_{(\overline{1},3,4,6,7)} = \langle\, \rangle)$
$\leadsto_{\mathit{SAT}}$	$F, [\overline{1}, \overline{5}], [\overline{1}] :: \mathit{SAT}$	

Relevant Clauses

- **Definition** A clause C is **relevant** in $F :: J$
  iff $C \in F$ and there exist I, L, I' such that $J = I, L, I'$ and $C|_I = [L]$
- **relevant**$(F :: J) = \{C \in F \mid C \text{ is relevant in } F :: J\}$

Conflict-Directed Backtracking and Learning

- **Replace naive backtracking by conflict-directed backtracking and learning**
- $F :: J, \dot{L}, J' \quad \leadsto_{CDBL} \quad F, D :: J, L'$ **iff**
  - **there exists** $C \in F$ **such that** $C|_{J,\dot{L},J'} = [\,]$
  - **there is a linear resolution derivation from** C **to** D **wrt relevant**$(F :: J, \dot{L}, J')$
  - $D|_J = [L']$

The DPLL-CDBL Calculus

- **Given a CNF-formula F**
- **The computation is initialized by $F :: (\,)$**
- **The rules of the calculus are**
  SAT, UNSAT, DECIDE, CDBL, TAUT, SUBS, UNIT and PURE
- **Computation terminates if**
  - **a node $F' ::$ SAT is reached in which case F is satisfiable or**
  - **a node $F' ::$ UNSAT is reached in which case F is unsatisfiable**

Another Example

$F :: (\,)$ where $F = \langle[\bar{1}, 2], [\bar{2}, 3], [\bar{4}, 5], [\bar{5}, 6], [\bar{7}, 8], [\bar{8}, 9], [\bar{3}, \bar{8}, \bar{9}]\rangle$

$\rightsquigarrow_{DECIDE}$ $F :: (\dot{1})$ $\quad (F|_{(1)} = \langle[2], [\bar{2}, 3], [\bar{4}, 5], [\bar{5}, 6], [\bar{7}, 8], [\bar{8}, 9], [\bar{3}, \bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F :: (\dot{1}, 2)$ $\quad (F|_{(1,2)} = \langle[3], [\bar{4}, 5], [\bar{5}, 6], [\bar{7}, 8], [\bar{8}, 9], [\bar{3}, \bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F :: (\dot{1}, 2, 3)$ $\quad (F|_{(\dot{1},2,3)} = \langle[\bar{4}, 5], [\bar{5}, 6], [\bar{7}, 8], [\bar{8}, 9], [\bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{DECIDE}$ $F :: (\dot{1}, 2, 3, \dot{4})$ $\quad (F|_{(\dot{1},2,3,\dot{4})} = \langle[5], [\bar{5}, 6], [\bar{7}, 8], [\bar{8}, 9], [\bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F :: (\dot{1}, 2, 3, \dot{4}, 5)$ $\quad (F|_{(\dot{1},2,3,\dot{4},5)} = \langle[6], [\bar{7}, 8], [\bar{8}, 9], [\bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F :: (\dot{1}, 2, 3, \dot{4}, 5, 6)$ $\quad (F|_{(\dot{1},2,3,\dot{4},5,6)} = \langle[\bar{7}, 8], [\bar{8}, 9], [\bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{DECIDE}$ $F :: (\dot{1}, 2, 3, \dot{4}, 5, 6, \dot{7})$ $\quad (F|_{(\dot{1},2,3,\dot{4},5,6,\dot{7})} = \langle[8], [\bar{8}, 9], [\bar{8}, \bar{9}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F :: (\dot{1}, 2, 3, \dot{4}, 5, 6, \dot{7}, 8)$ $\quad (F|_{(\dot{1},2,3,\dot{4},5,6,\dot{7},8)} = \langle[9], [\bar{9}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F :: (\dot{1}, 2, 3, \dot{4}, 5, 6, \dot{7}, 8, 9)$ $\quad (F|_{(\dot{1},2,3,\dot{4},5,6,\dot{7},8,9)} = \langle[\,]\rangle)$

$\rightsquigarrow_{CDBL}$ $F, [\bar{3}, \bar{8}] :: (\dot{1}, 2, 3, \bar{8})$ $\quad ((F, [\bar{3}, \bar{8}])|_{(\dot{1},2,3,\bar{8})} = \langle[\bar{4}, 5], [\bar{5}, 6], [\bar{7}]\rangle)$

$\rightsquigarrow_{UNIT}$ $F, [\bar{3}, \bar{8}] :: (\dot{1}, 2, 3, \bar{8}, \bar{7})$ $\quad ((F, [\bar{3}, \bar{8}])|_{(\dot{1},2,3,\bar{8},\bar{7})} = \langle[\bar{4}, 5], [\bar{5}, 6]\rangle)$

$\rightsquigarrow_{PURE}$ $F, [\bar{3}, \bar{8}] :: (\dot{1}, 2, 3, \bar{8}, \bar{7}, \bar{4})$ $\quad ((F, [\bar{3}, \bar{8}])|_{(\dot{1},2,3,\bar{8},\bar{7},\bar{4})} = \langle[\bar{5}, 6]\rangle)$

$\rightsquigarrow_{PURE}$ $F, [\bar{3}, \bar{8}] :: (\dot{1}, 2, 3, \bar{8}, \bar{7}, \bar{4}, \bar{5})$ $\quad ((F, [\bar{3}, \bar{8}])|_{(\dot{1},2,3,\bar{8},\bar{7},\bar{4},\bar{5})} = \langle\,\rangle)$

Another Example – Conflict Analysis

- relevant($F :: (\dot{1}, 2, 3, \dot{4}, 5, 6, \dot{7}, 8, 9)$) contains of the following clauses

$$
\begin{array}{lcll}
C_1 & = & [\overline{1}, 2] & ([\overline{1}, 2]|_{(1)} = [2]) \\
C_2 & = & [\overline{2}, 3] & ([\overline{2}, 3]|_{(1,2)} = [3]) \\
C_3 & = & [\overline{4}, 5] & ([\overline{4}, 5]|_{(1,2,3,4)} = [5]) \\
C_4 & = & [\overline{5}, 6] & ([\overline{5}, 6]|_{(1,2,3,4,5)} = [6]) \\
C_5 & = & [\overline{7}, 8] & ([\overline{7}, 8]|_{(1,2,3,4,5,6,7)} = [8]) \\
C_6 & = & [\overline{8}, 9] & ([\overline{8}, 9]|_{(1,2,3,4,5,6,,7,8)} = [9])
\end{array}
$$

- The conflict clause is $C = [\overline{3}, \overline{8}, \overline{9}]$
- We obtain the following linear derivation from C wrt $\{C_i \mid 1 \leq i \leq 6\}$

$$
\begin{array}{lcll}
C_7 & = & [\overline{8}, \overline{3}] & (\text{res}(C, C_6)) \\
C_8 & = & [\overline{7}, \overline{3}] & (\text{res}(C_7, C_5)) \\
C_9 & = & [\overline{7}, \overline{2}] & (\text{res}(C_8, C_2)) \\
C_{10} & = & [\overline{7}, \overline{1}] & (\text{res}(C_9, C_1))
\end{array}
$$

- In principle, all derived clauses could have been added!

Another Example – Continued

- There is an alternative application of CDBL

$$
\begin{array}{lll}
F :: (\,) \text{ where } F = \langle[\overline{1}, 2], [\overline{2}, 3], [\overline{4}, 5], [\overline{5}, 6], [\overline{7}, 8], [\overline{8}, 9], [\overline{3}, \overline{8}, \overline{9}]\rangle & & \\
\leadsto_{\mathit{DECIDE}} & F :: (\dot{\overline{1}}) & (F|_{(\overline{1})} = \langle[2], [\overline{2}, 3], [\overline{4}, 5], [\overline{5}, 6], [\overline{7}, 8], [\overline{8}, 9], [\overline{3}, \overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F :: (\dot{\overline{1}}, 2) & (F|_{(\overline{1},2)} = \langle[3], [\overline{4}, 5], [\overline{5}, 6], [\overline{7}, 8], [\overline{8}, 9], [\overline{3}, \overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F :: (\dot{\overline{1}}, 2, 3) & (F|_{(\dot{\overline{1}},2,3)} = \langle[\overline{4}, 5], [\overline{5}, 6], [\overline{7}, 8], [\overline{8}, 9], [\overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{DECIDE}} & F :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}) & (F|_{(\dot{\overline{1}},2,3,\dot{\overline{4}})} = \langle[5], [\overline{5}, 6], [\overline{7}, 8], [\overline{8}, 9], [\overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5) & (F|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5)} = \langle[6], [\overline{7}, 8], [\overline{8}, 9], [\overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5, 6) & (F|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5,6)} = \langle[\overline{7}, 8], [\overline{8}, 9], [\overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{DECIDE}} & F :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5, 6, \dot{\overline{7}}) & (F|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5,6,\dot{\overline{7}})} = \langle[8], [\overline{8}, 9], [\overline{8}, \overline{9}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5, 6, \dot{\overline{7}}, 8) & (F|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5,6,\dot{\overline{7}},8)} = \langle[9], [\overline{9}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5, 6, \dot{\overline{7}}, 8, 9) & (F|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5,6,\dot{\overline{7}},8,9)} = \langle[\,]\rangle) \\
\leadsto_{\mathit{CDBL}} & F, [\overline{3}, \overline{8}] :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5, 6, \overline{8}) & ((F, [\overline{3}, \overline{8}])|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5,6,\overline{8})} = \langle[\overline{7}]\rangle) \\
\leadsto_{\mathit{UNIT}} & F, [\overline{3}, \overline{8}] :: (\dot{\overline{1}}, 2, 3, \dot{\overline{4}}, 5, 6, \overline{8}, \overline{7}) & ((F, [\overline{3}, \overline{8}])|_{(\dot{\overline{1}},2,3,\dot{\overline{4}},5,6,\overline{8},\overline{7})} = \langle\,\rangle)
\end{array}
$$

DPLL-CDBL – Results

- **Theorem** DPLL-CDBL is terminating, sound and complete
- **Proof**
  - Termination and soundness follow from corresponding results for the GenericCDCL calculus, which will be presented later in the lecture
  - Completeness: to do

GenericCDCL

- H., Manthey, Philipp, Steinke: GenericCDCL – A Formalization of Modern Propositional Satisfiability Solvers. In: Proc. POS-14, Le Berre (ed.), EPiC Series 27, 89-102: 2014, EasyChair, http://www.easychair.org

- F and F' are **equisatisfiable**, in symbols $F \equiv_{SAT} F'$,
  iff either both are satisfiable or both are unsatisfiable

The Rules of the GenericCDCL Calculus

$F :: J$	$\leadsto_{SAT}$	SAT	iff	$F\|_J = \langle\,\rangle$
$F :: J$	$\leadsto_{UNSAT}$	UNSAT	iff	$[\,] \in F\|_J$ and J contains only propagation variables
$F :: J$	$\leadsto_{DEC}$	$F :: J, \dot{L}$	iff	$L \in \textbf{atoms}(F) \cup \overline{\textbf{atoms}(F)}$ and $\{L, \overline{L}\} \cap J = \emptyset$
$F :: J$	$\leadsto_{INF}$	$F :: J, L$	iff	$F\|_J \equiv_{SAT} F\|_{J,L}$, $L \in \textbf{atoms}(F) \cup \overline{\textbf{atoms}(F)}$ and $\{L, \overline{L}\} \cap J = \emptyset$
$F :: J$	$\leadsto_{LEARN}$	$F, C :: J$	iff	$F \models C$
$F :: J$	$\leadsto_{REMOVE}$	$F \setminus \{C\} :: J$	iff	$F \setminus \{C\} \models C$
$F :: J, J'$	$\leadsto_{BACK}$	$F :: J$		
$F :: (\,)$	$\leadsto_{INP}$	$F' :: (\,)$	iff	$F \equiv_{SAT} F'$

Some Comments

- The Rules
  - **SAT UNSAT DEC** minor changes
  - **INF** covers UNIT and PURE as well as many other techniques
  - **LEARN** covers all learning techniques in SAT solvers
  - **REMOVE** covers SUBS as well as TAUT, but also allows to remove previously learned clauses if they are not effective
  - **BACK** covers naive backtracking, backjumping as well as restarts
  - **INP** allows the application of all simplification techniques
- GenericCDCL covers all systematic SAT solvers
- 'All' is to be understood as 'to the best of our knowledge, all'

The GenericCDCL Calculus

- **Given a CNF-formula F**
- **The computation is initialized by $F :: (\,)$**
- **The rules of the calculus are SAT, UNSAT, DEC, INF, LEARN, REMOVE, BACK and INP**
- **Computation terminates if**
  - **the node SAT is reached in which case F is satisfiable or**
  - **the node UNSAT is reached in whch case F is unsatisfiable**

Invariants

- **Proposition** If $F :: (\,) \overset{n}{\leadsto} G :: J$, then
  - $F \equiv_{SAT} G$
  - $G|_{J_1} \equiv_{SAT} G|_{J_1,L}$, for all J_1, J_2 and propagation literal L with $J = J_1, L, J_2$
- **Proof** by induction on n
- **Exercise** complete the proof

Soundness

- **Theorem** If $F :: (\,) \overset{*}{\leadsto} G :: J \leadsto_{SAT}$ SAT, then F is satisfiable
  If $F :: (\,) \overset{*}{\leadsto} G :: J \leadsto_{UNSAT}$ UNSAT, then F is unsatisfiable
- **Proof** follows immediately from the previous proposition
- **Exercise** complete the proof

Completeness

- **Theorem** If F is satisfiable, then $F :: (\,) \overset{*}{\leadsto}$ SAT
  If F is unsatisfiable, then $F :: (\,) \overset{*}{\leadsto}$ UNSAT
- **Proof**
  - Suppose F is satisfiable
    Then we find a model $J = (L_1, \ldots, L_n)$ for F
    Then $F :: (\,) \overset{n}{\leadsto}_{DEC} F :: (\dot{L}_1, \ldots, \dot{L}_n) \leadsto_{SAT}$ SAT
  - Suppose F is unsatisfiable
    Then $F \models [\,]$
    Then $F :: (\,) \leadsto_{LEARN} F, [\,] :: (\,) \leadsto_{UNSAT}$ UNSAT

Confluence for Reachable States

- **Theorem** If $F :: (\,) \overset{*}{\leadsto} T_1$ and $F :: (\,) \overset{*}{\leadsto} T_2$,
  then there exists T with $T_1 \overset{*}{\leadsto} T$ and $T_2 \overset{*}{\leadsto} T$
- **Proof** follows from the completeness of GenericCDCL
  and its ability to perform restarts with the help of the BACK rule
- **Exercise** complete the proof

Termination Analysis

- GenericCDCL does not terminate due to
  - possibly infinite sequences of LEARN and REMOVE
  - possibly infinite sequences of restarts
  - possibly infinite sequences of INP
- Fairness Criteria
  - Each clause $C \subseteq \text{lits}(F)$ is learned at most finitely many times
    - Eventually, LEARN is no longer applicable
  - The number of restarts and the number of applications of INP is bounded
    - Eventually, restarts and INP are no longer applicable
- Alternative fairness criteria are possible